ALget Common Use

This helper class allows retrieval of any value related to the record or user. It takes the passed XML file and parses the data structure so class methods can access it.

The discussion here assumes that “get_data” has been defined in the main function as follows:

```
get_data = ALget(inputXMLFile)
```

Retrieving Values

The specific examples call functions of “get_data” to retrieve values. Many types of values can be retrieved from the input file.

A standard field in the current record

The `value` function retrieves the content from a standard field in the record.

Syntax: `value(fieldName, recordName)`

Value returned: depends on field domain

Example

```
# Get the value in the 'type_of_problem' field. This uses the name of the field,
not the label.
        Value1 = get_data.value('type_of_problem')
```

Attached files from the current record

The `attachedFiles` function retrieves attachment data from a specified File with Versioning field in the record. Note that it does not work with the File field type.

Syntax: `attachedFiles(fieldName, recordName)`

Value returned: list of ALAttachedFile objects

Example

```
# Get information about attached files. The files are retrieved as a list of
objects with attributes for name and path.
# These are the only attributes available.
        Value2 = get_data.attachedFiles('attached_files')
        Name1 = Value2[0].name
        Path1 = Value2[0].path
```

A field related to the user or other global variable

The `globalVariable` function returns the value of a user-related or KB global variable, or returns the default value if the variable is absent or null. The `globalVariablesNamesList` function can be used to find all available variable names.

Syntax: `globalVariable(variableName, default = None)`

Value returned: depends on variable domain

Example

```
# Get the name of the user who triggered the script.
        Value3 = get_data.globalVariable('my_full_name')
```

A field from a linked record

The `linkedValue` function retrieves the linkedRecordFieldName field value for the linkedRecordIdx donor record that is linked to the recordName record in the fieldName field. So, if you want to use a Contract record to pull data from the assigned Person record, you would use the Contract record in recordName, the Contract's linked field as fieldName, the Person record in linkedRecordIdx, and the data field in the Person record as linkedRecordFieldName.

Syntax: `linkedValue(fieldName, linkedRecordIdx, linkedRecordFieldName, recordName)`

Value returned: depends on donor field domain

A multiple linked field in the current record

The `valueMultipleLF` function retrieves the values from the donor record, if the number of values is relatively small. The XML retrieved by this function is limited by the maximum text field size (max_text_field_size_in_db global variable), so if there are hundreds of records with large field values, this function might not retrieve all of the data. If you need to iterate over all the linked records in these cases, use the `linkedRecordsNamesList` and `linkedValue` functions.

Syntax: `valueMultipleLF(fieldName, recordName)`

Value returned: ALMultipleLFContainer object

Example

```
# Get information about the affected departments, which is a Link to a Single Field
with multiple values enabled.
# This creates an object with built-in functions for checking contained values and
finding records containing a particular value.
#valueMultipleLF is available for all types of linked field sets when multiple
records are allowed, but not related tables or embedded search results.
        Value5 = get_data.valueMultipleLF('affected_departments')
        if Value5.containsValue("Finance")...
                NextID = Value5.findByValue("Finance")


# Retrieve information about the 'assignee' field, which is a link to both the
"teams" and "contacts" tables. This creates a list of objects.
# Then check if the 'assignee' field has a team selected instead of a contact.
        Value6 = get_data.linkedRecordsNamesList("assigned_team")
        if get_data.tableNameForRecord(Value6[0]) == "teams":...
```

Objects Returned

In addition to integer and text values, some ALget functions return special objects.

ALAttachedFile

This object presents an attached file with its file name (`name`) and location (`path`).

ALMultipleLFContainer

This object represents multiple imported values for a linked field. There are several available functions for this object type.

- `findByValue(value)`: Input a value to search for and receive a [recordID, value] pair if it exists, or None otherwise.
- `containsValue(value)`: Input a value to search for and receive True if the object contains any lines with the value.
- `items()`: Receive a list of all [recordID, value] pairs in the container.
- `itemsMap()`: Receive a dict of { donorRecordId: value, ..., donorRecordX: valueX } for the object.